Pending Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1-12. (Canceled)

13. (Currently amended): A method, comprising:

at a processing engine within a processor having multiple processing engines, individual ones of the engines including corresponding arbiters to select threads for execution,

executing at least one instruction of a first thread having a first program counter, the at least one instruction including at least one instruction to issue a request to a resource shared by the multiple processing engines;

executing one or more additional instructions of the first thread, after executing the at least one instruction to issue the request to the shared resource;

swapping execution to a second thread having a second program counter after processing engine execution of the at least one instruction to issue the request to the shared resource; and

swapping execution to the first thread after detection of a signal generated in response to the request to the shared resource.

14. (Previously presented): The method of claim 13, further comprising selecting a thread to execute by the processing engine.

15. (Previously presented): The method of claim 14,

wherein threads of the processing engine comprise threads having one of the following states:

currently being executed by the engine;

available for execution, but not currently executing;

waiting for detection of a signal before being available for execution; and

wherein the selecting comprises selecting a thread from among threads available for execution, but not currently executing.

16. (Previously presented): The method of claim 15, wherein the selecting the thread comprises selecting the thread based on a round-robin among the threads available for execution.

17. (Previously presented): The method of claim 14, wherein selecting the thread comprises selecting a thread other than the first thread after detection of the signal and before swapping execution to the first thread.

18. (Previously presented): The method of claim 13, wherein swapping execution comprises selecting a program counter associated with a selected thread.

19. (Cancelled)

20. (Previously presented): The method of claim 13, further comprising

executing an instruction of the first thread explicitly requesting thread swapping; and

swapping execution to the second thread in response to the instruction explicitly requesting thread swapping.

21. (Previously presented): The method of claim 20, wherein the instruction of the first thread explicitly requesting thread swapping does not comprise an instruction to issue a request to a shared resource.

22. (Previously presented): The method of claim 13, wherein the at least one instruction identifies the signal.

23. (Previously presented): The method of claim 13, wherein the signal comprises a signal generated in response to servicing of the request.

24. (Previously presented): The method of claim 13, wherein the shared resource comprises one of the following: a memory shared by the multiple processing engines internal to the processor and a memory shared by the multiple processing engines external to the processor.

25. (Previously presented): The method of claim 13, further comprising:

receiving a packet; and

processing the packet using the first thread.

26. (Currently amended): A network device, comprising:

at least one Ethernet media access controller; and

at least one network processor communicatively coupled to the at least one Ethernet media access controller, the at least one network processor comprising:

multiple, multi-threaded processing engines, individual ones of the engines including an arbiter to select a thread to execute, the multiple multi-threaded processing engines configured to execute one or more instructions of a first thread, after execution of an instruction of the first thread that issued a request to a shared resource and before swapping of the first thread for another thread;

a memory internal to the network processor shared by the multiple processing engines;

at least one interface to at least one memory external to the network processor; and

at least one interface to the at least one Ethernet media access controller.

27. (Previously presented): The device of claim 26,

wherein threads of the individual ones of the processing engines comprise threads having one of the following states:

currently being executed by the processing engine;

available for execution, but not currently executed by the processing engine;

waiting for detection of a signal associated with a request to a resource shared by the processing engines before being available for execution; and

wherein the arbiter of an individual processing engine selects a thread from among threads available for execution, but not currently executing.

28. (Previously presented) The device of claim 27, wherein the arbiter selects based on a round-robin among the threads available for execution.

29. (Previously presented): The device of claim 26, wherein the individual processing engines use a program counter associated with the thread selected by the processing engine's arbiter.

30. (Previously presented): The device of claim 26, wherein the processing engines feature an instruction set that includes at least one instruction explicitly requesting a currently thread swap.

31. (Currently amended): A network processor comprising:

multiple, multi-threaded processing engines, individual ones of the engines including an arbiter to select a thread to execute, the multiple multi-threaded processing engines configured to execute one or more instructions of a first thread, after execution of an instruction of the first thread that issued a request to a shared resource and before swapping of the first thread for another thread;

a memory internal to the network processor shared by the multiple processing engines;

at least one interface to at least one memory external to the network processor; and

at least one interface to at least one media access controller.

32. (Previously presented): The network processor of claim 31,

wherein threads of individual ones of the processing engines comprise threads having one of the following states:

currently being executed by the processing engine;

available for execution, but not currently executed by the processing engine;

waiting for detection of a signal associated with a request to a resource shared by the processing engines before being available for execution; and

wherein the arbiter of the processing engine selects a thread from among threads available for execution, but not currently executing.

33. (Previously presented) The network processor of claim 32, wherein the arbiter selects based on a round-robin among the threads available for execution.

34. (Previously presented): The network processor of claim 31, wherein the processing engines use a program counter associated with the thread selected by the processing engine's arbiter.

35. (Previously presented): The network processor of claim 31, wherein the processing engines feature an instruction set that includes at least one instruction explicitly requesting a thread swap.